PC CONTROLLED HELICOPTER

MAIN PROJECT REPORT

Submitted by

BINOY P.B

LIXON DAVIS M

NEEMA MERLIN RODRIGUES

PRABIN K

in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY (B.TECH)

in

COMPUTER SCIENCE & ENGINEERING

of

UNIVERSITY OF CALICUT

Under the guidance of

Mr THOMAS GEORGE K

JUNE 2011

Department of Computer Science & Engineering
JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY
THRISSUR 679 531

Department of Computer Science & Engineering
JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY
THRISSUR 679 531

JUNE 2011

BONAFIDE CERTIFICATE

Certified that this project report **“ . . . PC CONTROLLED HELICOPTER . . . ”** being submitted in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** of **University of Calicut** is the bonafide work of **“ . . . BINOY P.B, LIXON DAVIS M, NEEMA MERLIN RODRIGUES, PRABIN K . . . ”**, who carried out the project work under our supervision.

Prof. Muralee Krishnan C
HOD
Dept. of CSE

Mr. Thomas George K
PROJECT GUIDE
Asst. Professor
Dept. of CSE

CONTENTS

ACKNOWLEDGEMENT

We take this opportunity to express our heartfelt gratitude to all respected personalities who had guided, inspired and helped us in the successful completion of this project.

First and foremost, we express our thanks to **The Lord Almighty** for guiding us in this endeavour and making it a success.

We are thankful to our Principal **Dr. U Lazar John** and the Management for providing us with excellent lab and infrastructure facilities.

Our sincere thanks to the Head of the Department of Computer Science & Engineering, Prof. **Muralee Krishnan C** for his valuable guidance and suggestions.

We would like to express our deepest gratitude to **Mr. Thomas George K** for his valuable contributions and guidance.

Last but not least, we thank all our teaching and non teaching staffs of Department of Computer Science & Engineering, and also our friends for their immense support and help in all the stages for the development of the project.

ABSTRACT

Our project is to develop a flying robot such as a helicopter which can be controlled from a PC or a laptop. Actual implementation is not possible in this project period and also not economical. So here we are planning to develop a small model of a helicopter and produces control from the PC.

There will be some auto pilot workings for making the stability in the robots motion. There are also some extra operations such as checking the battery backup and making auto landing operation in low battery condition, checking the communication and perform some auto returning operation to the base if failure occurred, perform some task for avoiding collision with object in its path of fly. The entire system works on some real time concerns for sudden actions in its path, communication, etc.

List of Figures

List of Tables

CHAPTER 1
Introduction

1.1 Overview

Robotics is the engineering science and technology of robots, and their design, manufacture, application, and structural disposition. Robotics is related to electronics, mechanics, and software. The word robot was introduced to the public by Czech writer Karel Capek in his play R.U.R. (Rossums Universal Robots), published in 1920. The term robotics was coined by Isaac Asimov in his 1941 science fiction short-story Liar!.

Here we are developing a small model of a helicopter which is controlled from the PC. Flying RC helicopter is really very exhilarating. Their versatility gives a RC pilot a complete access to the three-dimensional space in such a way that no other machines can!The project mainly consists of two parts. One is the actual flying body and the other is the control unit in the PC side. The flying body consist of the modules for receiving the signals from PC and the modules to produce control signals for the motors for moving purpose. There is another module which is for the vision. This module is rotatable so that view around the robot can be seen without the robots rotation.

The control unit in PC side consists of the transceivers for transmitting the control signals from the PC and also receive the status from the flying module. It also has some modules for PC interfacing, control unit and transceiver unit also.

1.2 Motivation and Technical Relevance

The existing flying robots are to be controlled using remote by viewing them. Our helicopter can be controlled from a base station according to the videos from the camera on the helicopter which we receive in the base station computer.

The following are the technical relevance and application areas of the project: in military application,take videos inside stadium,rescue operation,in flooded areas and as a spy application.

1.3 Progress of project

An Alloy Shark R/C helicopter was purchased and flying practice was done. The helicopter has the specifications such as intelligentized R/C system, full scale remote control, 360 degree exact directional, smooth hang performance, newly designed electricity saving function, safeguard battery model to prolong the user life. New helicopter models were designed and tested. A suitable model was found out and implemented.

The circuits for the project were designed. The hardware and software requirements were identified. Through the literature survey, the existing systems were learned and according to that the project plan was prepared.

The compact circuit for the flying part was designed and implemented. The circuit of base station was also implemented. The program was done to rotate the motor to left and right and to stop in Hi Tech C. There are modifications to be done.

The user interface for the above said operations have been built in Visual Basic. It has to be checked and implemented.

1.4 Member roles and responsibilities

The organizing function of management deals with devising roles for individuals and assigning responsibility for accomplishing project goals. Organization is basically motivated by the need for cooperation when the goals are not achievable by a single individual in a reasonable amount of time. We follow a decentralized-control team organization, i.e. all work is considered group work. The following table shows the team roles and responsibilities.

1.1: Team Organization

Name	Role/Responsibility
Prabin K	Leader
Binoy P.B	Designer
Neema Merlin Rodrigues	Designer
Lixon Davis M	Programmer

Prabin: Coordinates and controls all sorts of development of the project, Handles the flying part of project.

Binoy: Handles the base station of the project.

Lixon: Handles the code development for the project.

Neema: Handles the user interface for the project.

1.5 Layout

Chapter 2 presents the relevant documents referenced during the initial survey of the project concept.

Chapter 3 presents the details and features of the system proposed.

Chapter 4 includes the hardware and software requirements for the project.

Chapter 5 gives an overview of the schedule of the Interim/Term project work. Include member work effort and module allocations to each member here as per his/her responsibility.The section also presents the general architecture of our project concept. Various layers and players involved and the abstraction defined at each level or layer is mentioned.

Chapter 6 includes the algorithms or program code elements for the initial model (working implementation) of the project.

Chapter 7 includes the details of the unit tests, integration tests and proposals for future maintenance.

The last chapter, Chapter 8 summarizes the work done in this semester.

CHAPTER 2
Literature Survey

2.1 Documentation

For studying the area of robotics and our project of controlling the helicopter from PC by wireless transmission, we did a literature survey. The main papers which we went through are described in the next subsection.

2.1.1 Papers & related works

1.Wireless Computer Controlled Robotics Using PIC16F77 Microcontroller by Melonee Wise: This paper describes a project similar to ours. The main difference is that they use a finished robot and not a helicopter. There is no visual feedback too. But other hardware and software set up is explained here.

2.Introduction to PIC Programming by David Meiklejohn,Gooligum Electronics: This paper gave us an idea about PIC microcontroller and how to program it using Microchips PICkit2.

3.Programming a PIC Microcontroller-A Short Tutorial by Yesu Thommandru: This gave an idea of how to use MPLAB for programming PIC.

4.Control of an Autonomous Radio-Controlled Helicopter in a Modified Simulation Environment Using Proportional Integral Derivative Algorithms by Ainsmar X. Brown and Richard D. Garcia : This gave a brief idea of how the helicopters of army are controlled.

5.”The Caltech Helicopter Control Experiment”, Xiaoyun Zhu and Michiel van Nieuwstadt Control: This paper tells how to carry out experiments on helicopter models.

6.RCHeli: Infrastructure For PC-Controlled Micro Helicopter by Sanghoon Cha Brown University, scha@cs.brown.edu : This paper suggests helicopter control by PC, but it uses a Wiimote.

7.Computer based control system for a model helicopter : This is a report of semester project of a group in microengineering department which describes the difference between airplanes and helicopters and about their project on helicopter controlling as:”The thing is, helicopters are different from planes. An airplane by its very nature wants to fly and, if not interfered with too strongly by unusual events or by a deliberately incompetent pilot, it will fly. A helicopter does not want to fly. It is maintained in the air by a variety of forces and controls working in opposition to each other and, if there is any disturbance in this delicate balance, the helicopter stops flying; immediately and disastrously. There is no such thing as a gliding helicopter. The first big step in building an autonomous helicopter is obviously to develop a flying machine that is able to hover without human intervention.”

8.Work in Progress - Capstone Experience - Visual Navigation of an RC Vehicle using Wireless Video Feedback to a PC by Steven Northrup and Christopher Paros, Department of Electrical Engineering, Western New England College, Springfield, MA 01119, snorthru@ wnec.edu : This paper is related to our project in wireless video feedback to PC from a camera, though it uses a car model and parallel port interface instead of our helicopter and serial port interface.

9.Multi-Camera Visual Servoing of a Micro Helicopter Under Occlusions by Yuta Yoshihata, Kei Watanabe, Yasushi Iwatani and Koichi Hashimoto: Though we are not using multiple cameras or tracking objects, we got the following idea from this paper:”Autonomous control of unmanned helicopters has the advantage that there is no need to develop skilled workers and has potential for surveillance tasks in dangerous areas including forest-fire reconnaissance and monitoring of volcanic activity. For vehicle navigation, the use of computer vision as a sensor is effective in unmapped areas. Visual feedback control is also suitable for autonomous take-off and landings, since precise position control is required at a neighbourhood of the launch pad or the landing pad.”

10.The Design of a Wireless Data Acquisition and Transmission System by CAI Jun,YU Shun-Zheng and LIU Jing-li: This paper describes a low-powered and high-performance wireless data communication system, works in the ISM (Industrial Scientific Medical)Band which has been already successfully applied to Wireless vehicle system that is similar to ours. When the literature survey was conducted we came to know about the following. In all existing system, all documents are handled manually. So here there is a chance of getting documents corrupted. Also, in existing systems unauthorized access on server resource is possible. The authentication and authorization privileges provided are very less. The possibility of hacking of data when transmitted through the network is very high in the case of the existing systems. Privileges are given to various modules of the system.

CHAPTER 3

Proposed System

3.1 Development Plan

Here we use the evolutionary or incremental model or approach in our project. This is because there may be errors in earlier stages which we modify and only the final stage is delivered. This also helps in eliminating the flaws in the requirements analysis. Our project has a stepwise development.

Our mini project had the camera control system. But to mount it on a helicopter we modified the circuit and made it compact. In the prototype of the main project we had the compact circuit working. Also we flew the helicopter and from the feedback we understood the challenges in flying it. From these details we implemented the compact circuit on the helicopter which we designed and constructed. The controls were made from the PC using the user interface in VB. Using the keyboard better control of the helicopter is made possible.

The development plan and details of work according to that plan is shown in the figure below.

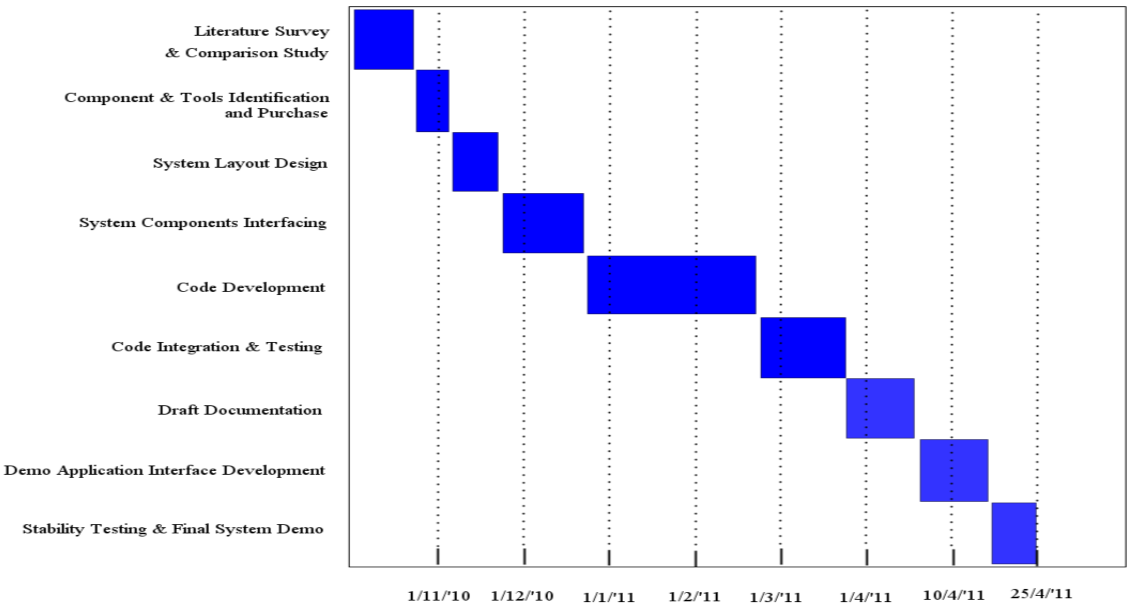

Fig. 3.1: Project plan

3.2 Conclusion

The evolutionary model can be used in the future also to modify the helicopter model for better performance. From the analysis of the present implementation, this can be done.

CHAPTER 4
System Requirements Specification

4.1 Software Requirements

Our project being related to many areas of engineering, has many hardware and software requirements.

4.1.1 Numbered list

The second common kind of list is *enumerated*.

1. Microsoft Windows XP
2. MPLAB IDE
3. Hi Tech C
4. MS Visual Basic 6.0
5. Express PCB
6. Express Schematic

4.2 Hardware Requirements

1. PIC16F877A Microprocessor with features as: High performance RISC CPU Operating speed: DC - 20 MHz clock input DC - 200 ns instruction cycle Up to 8K x 14 words of FLASH Program Memory, Up to 368 x 8 bytes of Data Memory (RAM) Up to 256 x 8 bytes of EEPROM Data Memory Interrupt capability (up to 14 sources)
2. Personal Computer
3. Stepper Motor
4. Driver ULN2803

5. Power supply
6. Encoder:HT12E
7. Decoder:HT12D
8. RF Camera
9. LM35 Temperature sensor
10. IR Sensor
11. MAX232 Level convertor
12. Buffer: SN74HCT541
13. Tuner Card

CHAPTER 5
Design & Analysis

5.1 System Analysis

The entire system is divided into 4 modules. The details of those modules and the responsibilities of each member of the group are described in the later sections.

5.1.1 Module breakup

5.1: Module Description

Module	Description
Flying Part	Helicopter with positionable camera unit
Base Station	Interface with PC
Coding	Brief Coding the Microcontroller
User Interface	Interface with user

5.1.2 Member effort

This section presents each member's effort in the team. The work-hours are also mention here alongside the module assigned.

5.2: Module Allocation

#	Task	Estimated Effort	Start Date	End Date	Person
1	Flying Part	(4 man-hrs)	(20/11/2010)	(14/04/2011)	Prabin
2	Base Station	(2 man-hrs)	(13/12/2010)	(14/04/2011)	binoy
3	Coding	(2 man-hrs)	(13/12/2010)	(20/04/2011)	Lixon
4	User Interface	(2 man-hrs)	(20/12/2010)	(04/04/2011)	Neema

The organizing function of management deals with devising roles for individuals and assigning responsibility for accomplishing project goals. Organization is basically motivated by the need for cooperation when the goals are not achievable by a single individual in a reasonable amount of time. We follow a decentralized-control team organization, i.e. all work is considered group work.

5.2 System Design

The architecture is the first high-level design of the system. It provides the medium for reasoning about and analyzing the global properties of the system, since global properties are determined not by individual components, but by the interaction of the whole set of components.

In designing the architecture, the designer must consider many functional requirements as well as many non-functional requirements, such as cost and reliability. Depending on some system requirements, some specific decomposition of the system into components and modes of interaction among those components is most appropriate.

The standard architectures are pipeline architecture, blackboard architecture, event-based architecture and domain-specific architecture. We follow the event-based architecture for our project. In this architecture, components respond to the occurrence of events. An event might be the detection of a signal by a sensor or the arrival of a message. Components are designed to create events or start their operations when they receive an event. Components wait for input from the environment. User interfaces are often structured to utilize mouse clicks or mouse drags as events and we use mouse clicks.

Our system receives the user inputs and the events of controlling the camera and the helicopter will occur according to that. The various sensors detect the changes in the environment and reports to the base station. The base station controls the helicopter using those data and they are the events in the system. Thus our system uses the event based architecture.

5.2.1 Use Case Models / Flow Diagrams

Every non-trivial engineering system must be specified. Control flow diagrams (CFDs) are a well-known and widely used notation for specifying the functions of an information system and how control flow from functions to functions. The CFDs of our system are given below.

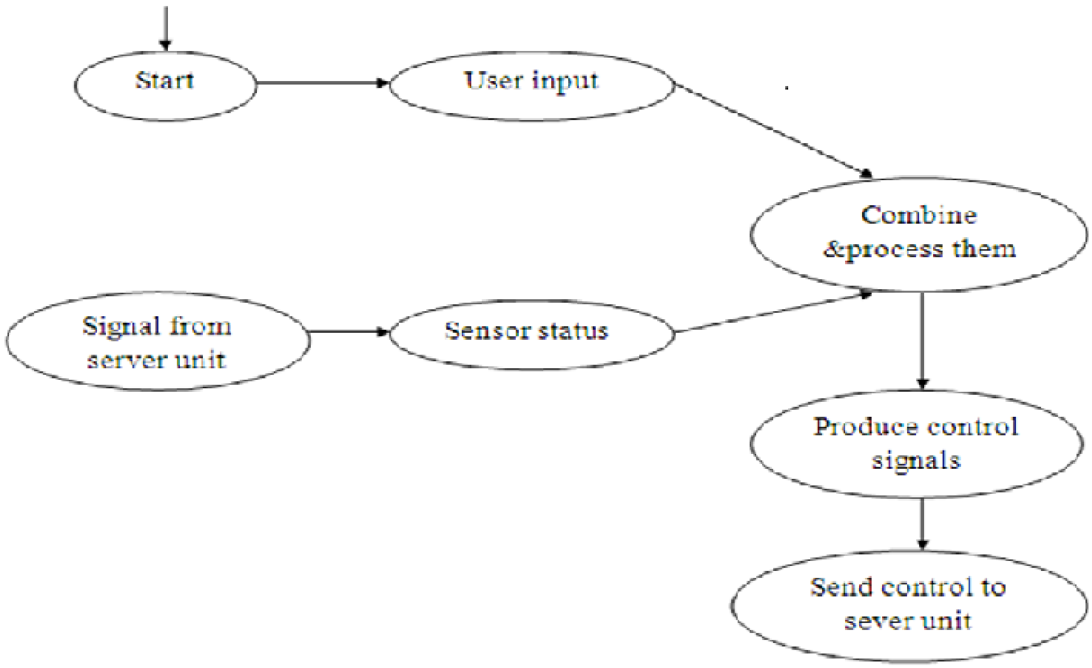

Fig. 5.1: Level 0 CFD User interface

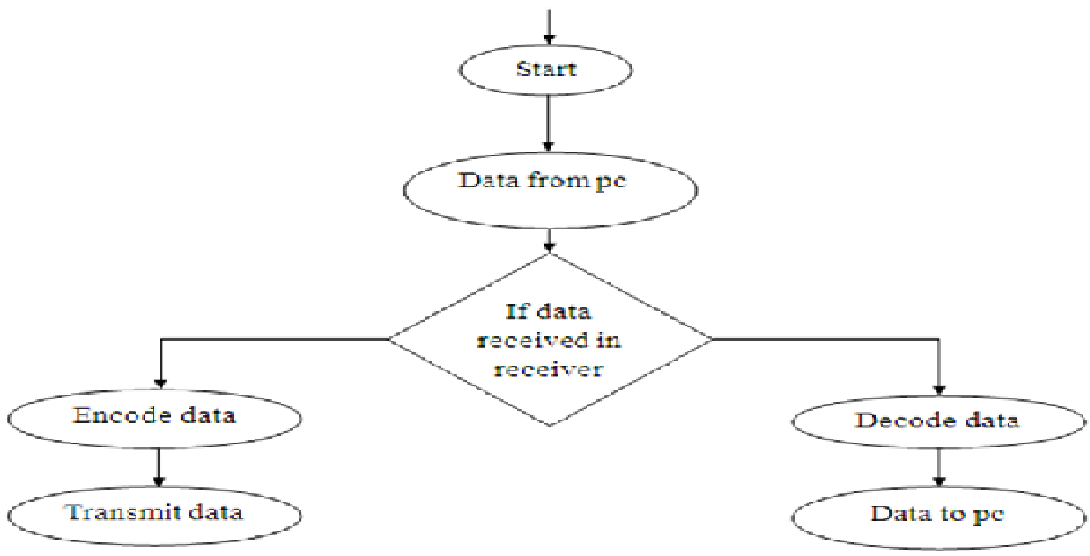

Fig. 5.2: Level 1 CFD Base station

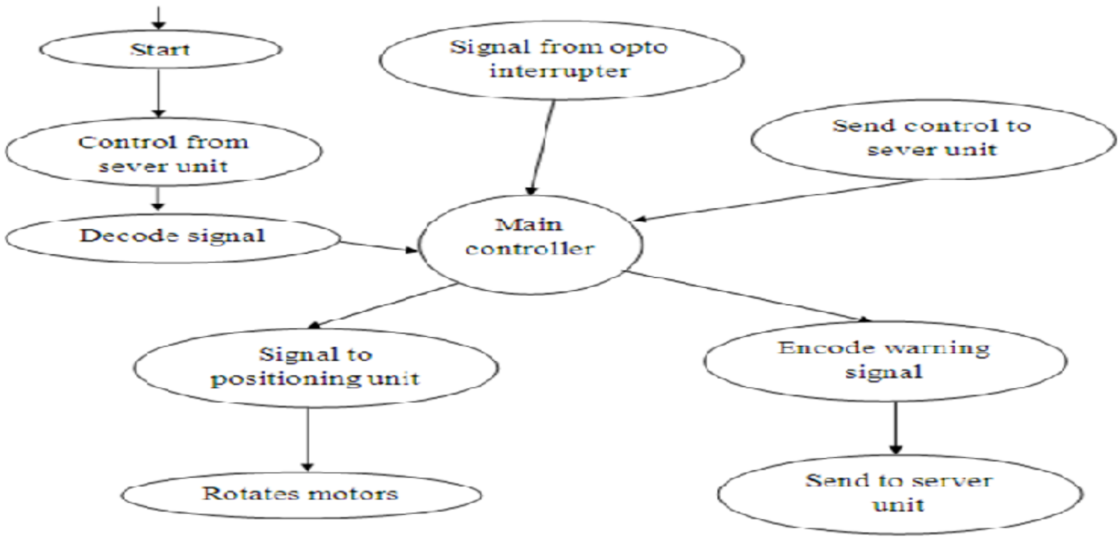

Fig. 5.3: Level 2 CFD Flying part

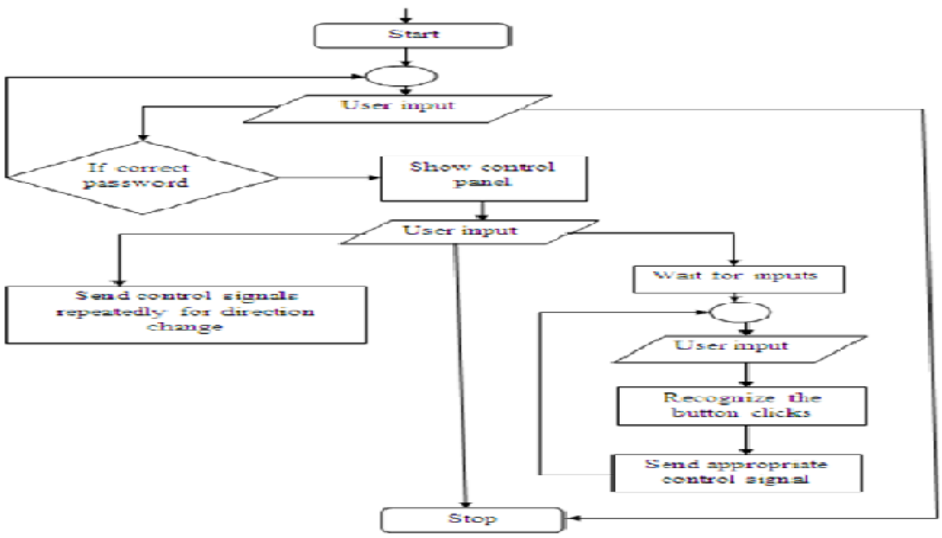

Fig. 5.4: Flow chart of entire system

The circuit diagrams which were designed for the project are given below:

Base Station

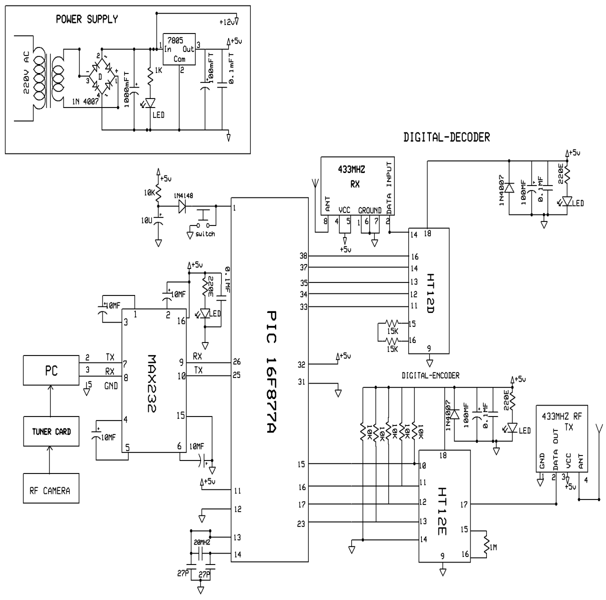

Fig. 5.5: PCKIT circuit diagram

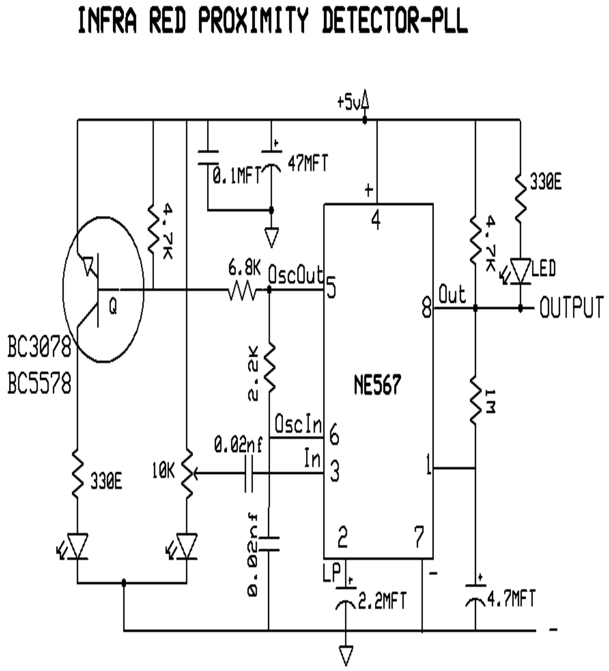

Fig. 5.6: IR proximity detector circuit diagram

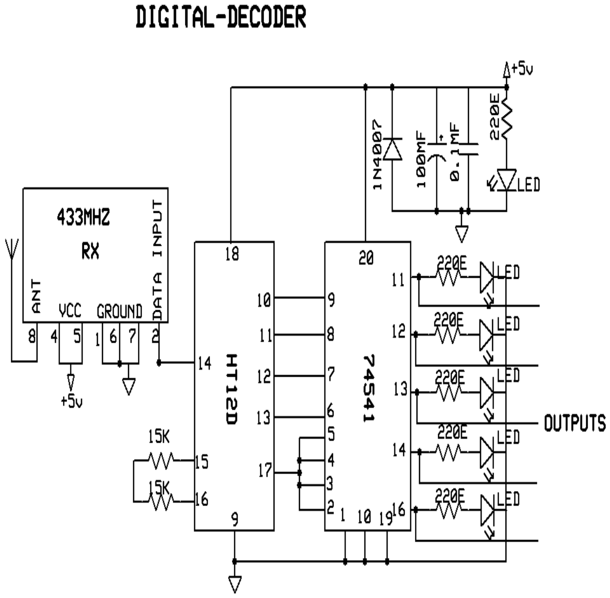

Fig. 5.7: Digital decoder circuit diagram

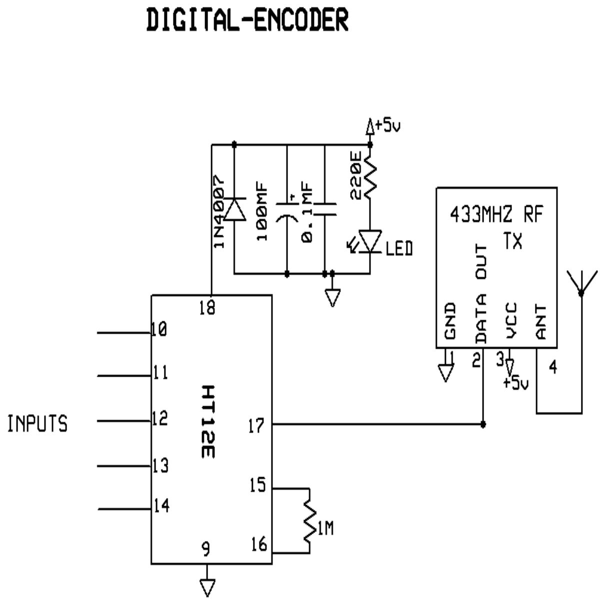

Fig. 5.8: Digital encoder circuit diagram

Flying part

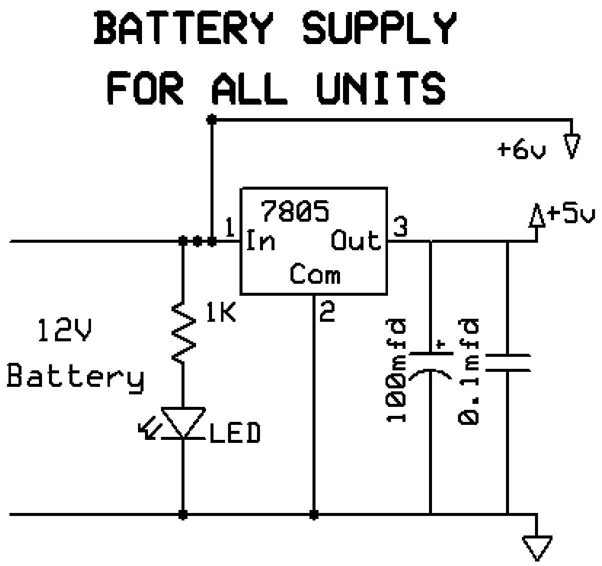

Fig. 5.9: Power supply circuit diagram

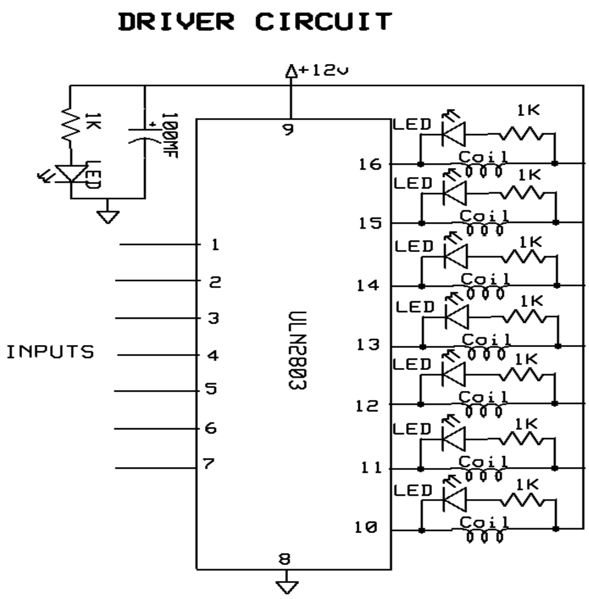

Fig. 5.10: Driver circuit diagram

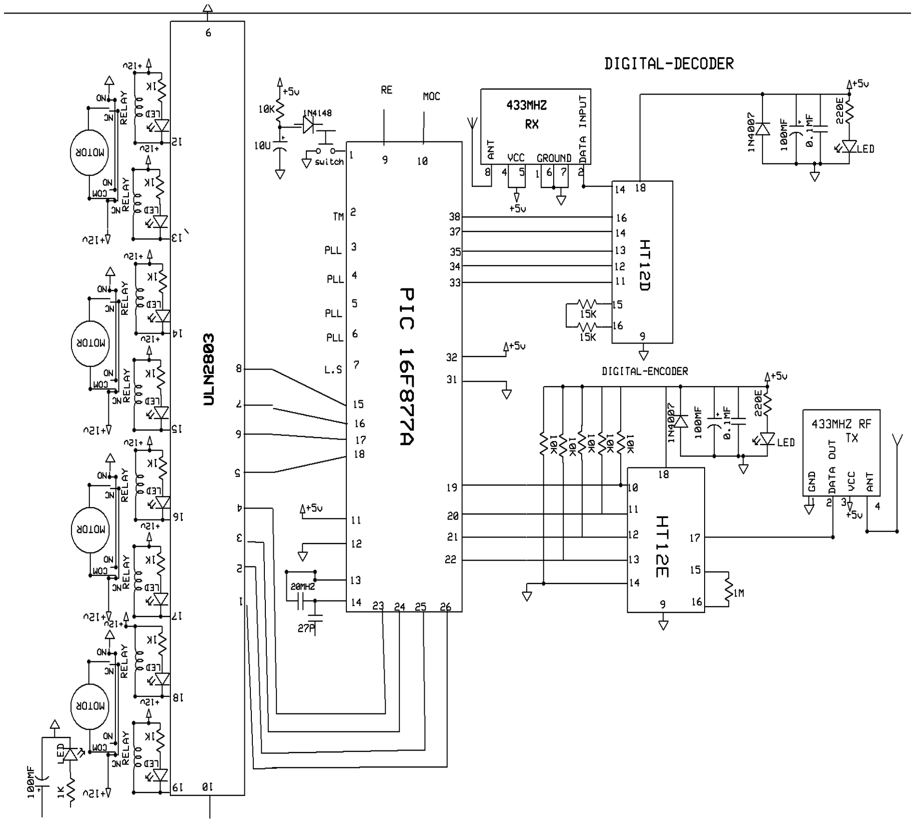

Fig. 5.11: PRI circuit diagram

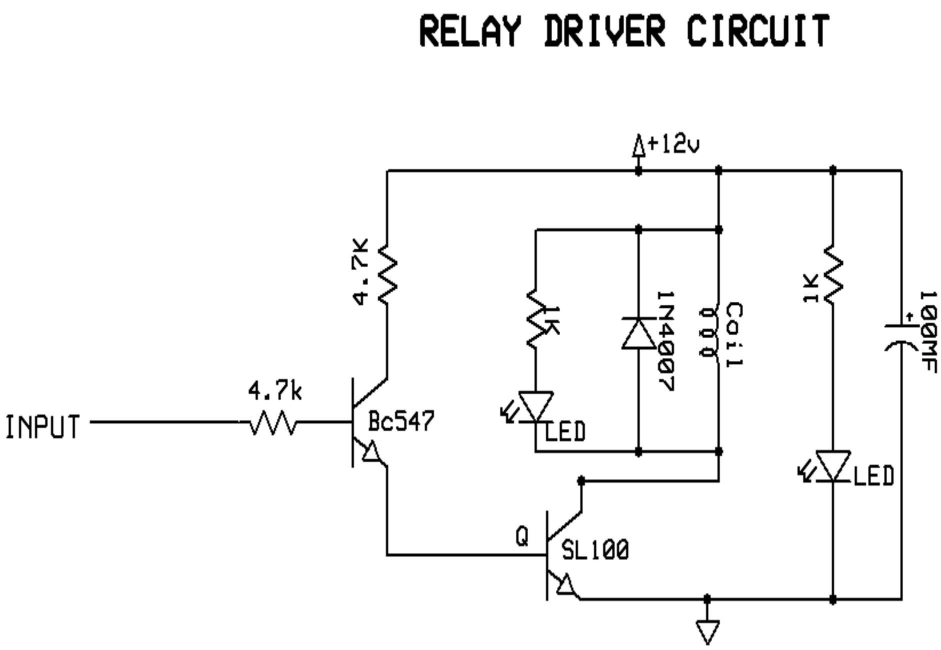

Fig. 5.12: Relay driver circuit diagram

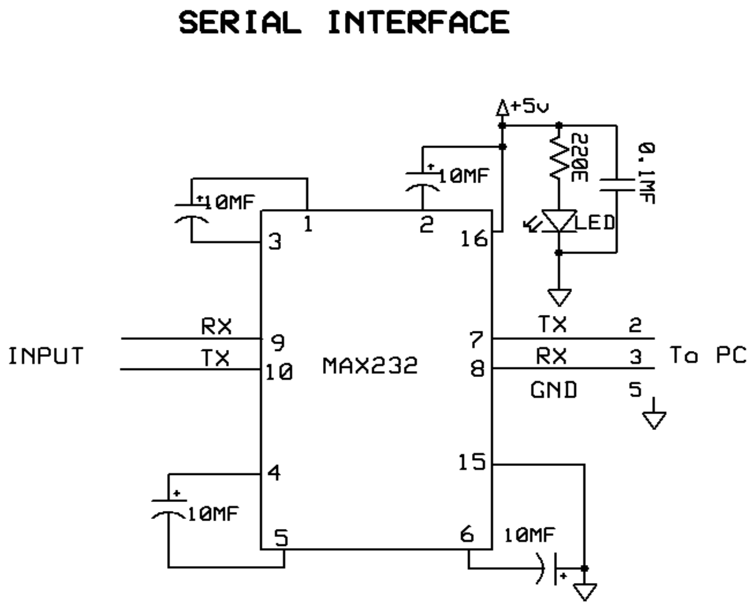

Fig. 5.13: Serial interface circuit diagram

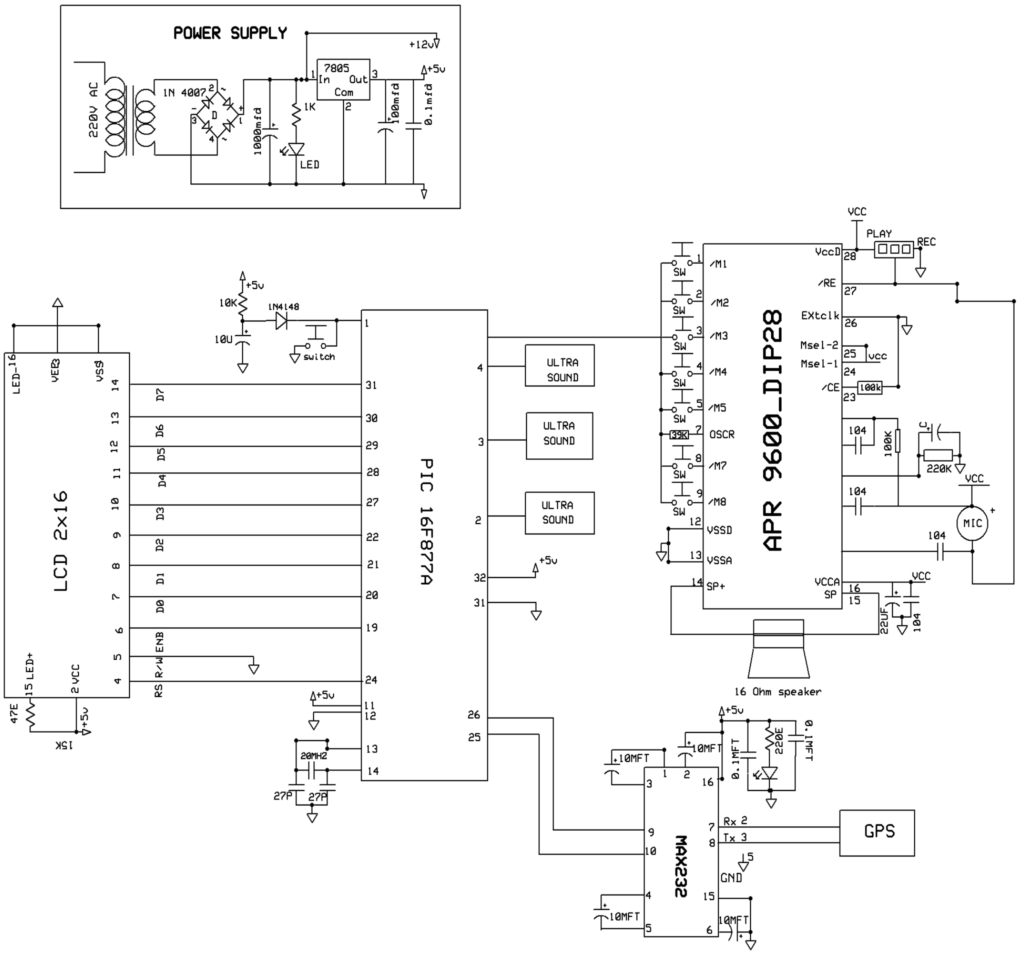

Fig. 5.14: TEA circuit diagram

CHAPTER 6

Implementation

6.1 Introduction

We are presenting the project with the operations of left and right rotations and to stop rotation of the motor and hence the camera on a helicopter which can also be controlled. This is done from the base station using the user interface in VB. The temporator and IR sensors for showing the direction are also included which shows the values in the user interface form. The video captured by the RF camera is also shown on the screen of the base station PC. The figure of the user interface is given below.

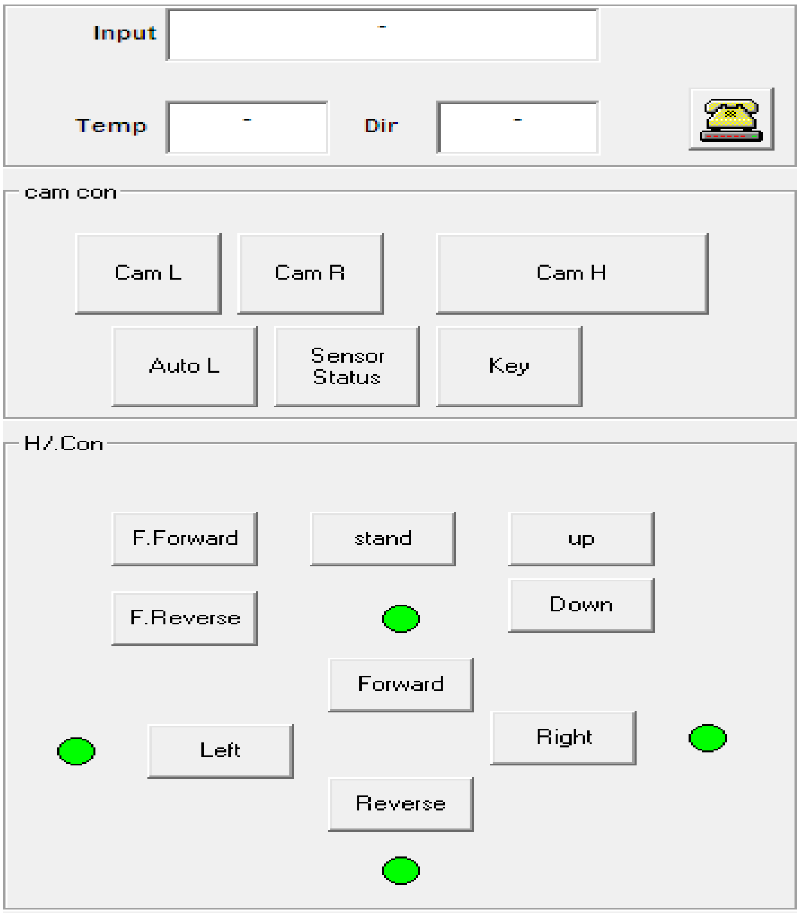

Fig. 6.1: User interface

6.2 Algorithm / Pseudo codes

User Interface

```
Private Sub cmdAutoL_Click()
 MSComm1.Output = ">"
End Sub
Private Sub cmdCamL_Click()
   MSComm1.Output = ";"
End Sub
Private Sub cmdCH_Click()
 MSComm1.Output = "="
End Sub
Private Sub cmdCr_Click()
 MSComm1.Output = "<"
End Sub
Private Sub cmdDown_Click()
 MSComm1.Output = "3"
End Sub
Private Sub cmdFFor_Click()
 MSComm1.Output = "8"
End Sub
Private Sub cmdFor_Click()
 MSComm1.Output = "4"
End Sub
Private Sub cmdFRev_Click()
 MSComm1.Output = "9"
End Sub
Private Sub cmdKey_KeyPress(KeyAscii As Integer)
  If KeyAscii = 53 Then cmdFor_Click
  If KeyAscii = 50 Then cmdRev_Click
  If KeyAscii = 49 Then cmdLeft_Click
  If KeyAscii = 51 Then cmdRight_Click
  If KeyAscii = 55 Then cmdFFor_Click
  If KeyAscii = 52 Then cmdFRev_Click
  If KeyAscii = 97 Then cmdCamL_Click
```

```
 If KeyAscii = 122 Then cmdCH_Click
 If KeyAscii = 115 Then cmdCr_Click
 If KeyAscii = 56 Then cmdstand_Click
 If KeyAscii = 57 Then cmdUP_Click
 If KeyAscii = 54 Then cmdDown_Click
 If KeyAscii = 98 Then cmdAutoL_Click
 If KeyAscii = 99 Then cmdSS_Click
End Sub
Private Sub cmdLeft_Click()
 MSComm1.Output = "7"
End Sub
Private Sub cmdRev_Click()
 MSComm1.Output = "5"
End Sub
Private Sub cmdRight_Click()
 MSComm1.Output = "6"
End Sub
Private Sub cmdSS_Click()
 MSComm1.Output = ":"
End Sub
Private Sub cmdstand_Click()
 MSComm1.Output = "1"
End Sub
Private Sub cmdUP_Click()
 MSComm1.Output = "2"
End Sub
Private Sub Form_Load()
 MSComm1.PortOpen = True
End Sub
Private Sub Form_Unload(Cancel As Integer)
 MSComm1.PortOpen = False
End Sub
Private Sub MSComm1_OnComm()
 Dim V As String
 Dim n As String
  txtInput = MSComm1.Input
  txtTemp = Mid(txtInput, 1, 3)
```

```
 txtDir = Mid(txtInput, 4, 1)
 V = txtDir
 If V = ">" Then
     Call ALL_GREEN
     Shape1.BackColor = vbRed
 ElseIf V = "=" Then
     Call ALL_GREEN
     Shape2.BackColor = vbRed
 ElseIf V = ";" Then
     Call ALL_GREEN
     Shape3.BackColor = vbRed
 ElseIf V = "7" Then
     Call ALL_GREEN
     Shape4.BackColor = vbRed
 End If
End Sub
Private Sub ALL_GREEN()
 Shape1.BackColor = vbGreen
 Shape2.BackColor = vbGreen
 Shape3.BackColor = vbGreen
 Shape4.BackColor = vbGreen
End Sub
```

Base Station

```
//XTAL -4Mhz Baudrate: 9600bps
#include<pic.h>
__CONFIG(HS & WDTDIS & PWRTEN & BOREN & UNPROTECT & LVPDIS);
static bit rs @((unsigned)&PORTE*8+0);
static bit en @((unsigned)&PORTE*8+1);
static bit rf_txen @((unsigned)&PORTA*8+4);
unsigned char i,j,ser=0x37,rf=0x00,portdata;
bit b;
//*************************************************
//        Function to introduce a delay
//*************************************************
void delay(unsigned int y)
```

```
{
while(y--);
}
//**************************************************
//        Function to send Command to LCD module
//**************************************************
void lcd_command(unsigned char com)
{
        PORTD=com;
        en=1;
        rs=0;
        delay(250);
        en=0;
        delay(250);
}
//**************************************************
//        Function to send Display data to LCD module
//**************************************************
void lcd_data(unsigned char dat)
{
        PORTD=dat;
        en=1;
        rs=1;
        delay(250);
        en=0;
        delay(250);
}
//*******************************************************
//   Function to send the display value from *word to Display
                                              data function
//*******************************************************
void lcd_condis(const unsigned char *word, unsigned int n)
{
        unsigned char i;
        for(i=0;i<n;i++)
        {
        lcd_data(word[i]);
```

```
        }
}
//*************************************************
//          LCD Initialization subroutine
//*************************************************
void lcd_init()
{
lcd_command(0x38);
lcd_command(0x06);
lcd_command(0x0c);
lcd_command(0x01);
lcd_command(0x80);
}
//*************************************************
//          Check Error
//*************************************************
void interrupt rcx(void)
{
if(RCIF==1)
{
RCIF=0;
ser=RCREG;
i=0x01;
}
if(INTF==1)
{
INTE=0;
GIE=0;
INTF=0;
rf=0x01;
INTE=1;
GIE=1;
}
}
//*************************************************
//          Main function
//*************************************************
```

```
void main()
{
ADCON1=0x07;
TRISE=0x00;
PORTE=0x00;
TRISA=0x00;
PORTA=0x00;
TRISB=0xff;
PORTB=0xff;
TRISC=0x80;
TRISD=0x00;
PORTC=0x80;
PORTD=0x00;
RBPU=0;
lcd_init();
delay(100);
lcd_condis("Aircraft Project",16);
i=0x00;
rf=0x00;
rf_txen = 1;
        SPBRG=0x19;
        BRGH=1;
        SYNC=0;
        CREN=1;
        TXEN=1;
        RCSTA=0x90;
        GIE=1;
        PEIE=1;
        RCIE=1;
        INTE=1;
        INTEDG=1;
        lcd_command(0xc0);
        delay(20000);
while(1)
{
if(i!=0x00)
{
```

```
lcd_command(0x80);
lcd_data(ser);
if((ser>=0x2a)||(ser<=0x7a))
{
                TXREG=ser;
                PORTA=ser;
                rf_txen = 0;
                i=0x00;
                delay(20600);
                delay(20600);
                rf_txen = 1;
}
}
if(rf!=0x00)
{
                lcd_command(0x87);
                rf=0x00;
                portdata =PORTB;
                portdata=(portdata>>4);
                portdata=(portdata&0x0f);
                portdata=(portdata+0x30);
                lcd_data(portdata);
                TXREG=portdata;
                delay(600);
}
}
}
```

Flying Part

```
//XTAL -4Mhz Baudrate: 9600bps
#include<pic.h>
__CONFIG(HS & WDTDIS & PWRTEN & BOREN & UNPROTECT & LVPDIS);
//ADC Obsdet- PORTA
//IR          - PORTE
//RF Rx       - PORTB
//RF Relay    - PORTC
```

```
//RF Tx          - PORTD (0-3)
//SM             - PORTD (4-7)
static bit rs @((unsigned)&PORTE*8+0);
static bit en @((unsigned)&PORTE*8+1);
static bit te @((unsigned)&PORTE*8+0);
static bit motor1_5v @((unsigned)&PORTC*8+0);
static bit motor1_6v @((unsigned)&PORTC*8+1);
static bit motor1_7v @((unsigned)&PORTC*8+2);
static bit motor2_5v @((unsigned)&PORTC*8+3);
static bit motor2_6v @((unsigned)&PORTC*8+4);
static bit motor2_7v @((unsigned)&PORTC*8+5);
static bit motor34_5v @((unsigned)&PORTC*8+6);
static bit motor34_6v @((unsigned)&PORTC*8+7);
static bit motor34_7v @((unsigned)&PORTE*8+1);
unsigned char i,j,ser=0x37,rf=0x00,portdata,sensorstatus=0x01;
bit b;
unsigned int v1,temp,adcvalue,count,smcount=0x00;
unsigned char sensor13,sensor12,sensor11,obsdet;
//*************************************************
//      Function to introduce a delay
//*************************************************
void delay(unsigned int y)
{
while(y--);
}
//*************************************************
//       Function to send Command to LCD module
//*************************************************
void lcd_command(unsigned char com)
{
/*      PORTD=com;
        en=1;
        rs=0;
        delay(250);
        en=0;
        delay(250); */
}
```

```
//*************************************************
//         Function to send Display data to LCD module
//*************************************************
void lcd_data(unsigned char dat)
{
/*         PORTD=dat;
           en=1;
           rs=1;
           delay(250);
           en=0;
           delay(250); */
}
//*******************************************************
//   Function to send the display value from *word to Display
                                                data function
//*******************************************************
void lcd_condis(const unsigned char *word, unsigned int n)
{
/*         unsigned char i;
           for(i=0;i<n;i++)
           {
           lcd_data(word[i]);
           } */
}
//*************************************************
//         LCD Initialization subroutine
//*************************************************
void lcd_init()
{
lcd_command(0x38);
lcd_command(0x06);
lcd_command(0x0c);
lcd_command(0x01);
lcd_command(0x80);
}
void adcconvert1()
{
```

```
CHS2=0;
CHS1=0;
CHS0=0;
delay(25);
ADON=1;
ADCON0=(ADCON0|0x04);
delay(100);
lcd_command(0x80);
temp=ADRESL/2;
adcvalue=ADRESH;
if (adcvalue!=0x00)
{
        if(adcvalue==0x01)
        {
        adcvalue=(0x0100+ADRESL)/0x02;
        temp=adcvalue;
        }
        else if(adcvalue==0x02)
        {
        adcvalue=(0x0200+ADRESL)/0x02;
        temp=adcvalue;
        }
        else if(adcvalue==0x03)
        {
        adcvalue=(0x0300+ADRESL)/0x02;
        temp=adcvalue;
        }
}
v1=temp/10;
v1=v1/10;
sensor13=(v1+0x30);
lcd_data(sensor13);

v1=temp/10;
v1=v1%10;
sensor12=(v1+0x30);
lcd_data(sensor12);
```

```
        v1=temp%10;
        sensor11=(v1+0x30);
        lcd_data(sensor11);
}
void standby()
{
motor1_5v=0;
motor1_6v=1;
motor1_7v=0;
motor2_5v=0;
motor2_6v=1;
motor2_7v=0;
motor34_5v=0;
motor34_6v=1;
motor34_7v=0;
}
void upward()
{
motor1_5v=0;
motor1_6v=0;
motor1_7v=1;
motor2_5v=0;
motor2_6v=0;
motor2_7v=1;
motor34_5v=0;
motor34_6v=0;
motor34_7v=1;
}
void downward()
{
motor1_5v=1;
motor1_6v=0;
motor1_7v=0;
motor2_5v=1;
motor2_6v=0;
motor2_7v=0;
```

```
motor34_5v=1;
motor34_6v=0;
motor34_7v=0;
}
void left()
{
motor1_5v=0;
motor1_6v=1;
motor1_7v=0;
motor2_5v=0;
motor2_6v=1;
motor2_7v=0;
motor34_5v=0;
motor34_6v=0;
motor34_7v=1;
}
void right()
{
motor1_5v=0;
motor1_6v=1;
motor1_7v=0;
motor2_5v=0;
motor2_6v=1;
motor2_7v=0;
motor34_5v=1;
motor34_6v=0;
motor34_7v=0;
}
void forward()
{
motor1_5v=0;
motor1_6v=0;
motor1_7v=1;
motor2_5v=1;
motor2_6v=0;
motor2_7v=0;
motor34_5v=0;
```

```
motor34_6v=1;
motor34_7v=0;
}
void reverse()
{
motor1_5v=1;
motor1_6v=0;
motor1_7v=0;
motor2_5v=0;
motor2_6v=0;
motor2_7v=1;
motor34_5v=0;
motor34_6v=1;
motor34_7v=0;
}
void fastforward()
{
motor1_5v=0;
motor1_6v=0;
motor1_7v=1;
motor2_5v=1;
motor2_6v=0;
motor2_7v=0;
motor34_5v=0;
motor34_6v=1;
motor34_7v=0;
}
void fastreverse()
{
motor1_5v=1;
motor1_6v=0;
motor1_7v=0;
motor2_5v=0;
motor2_6v=0;
motor2_7v=1;
motor34_5v=0;
motor34_6v=1;
```

```
motor34_7v=0;
}
void obstacledetection_chkright()
{
            if(RA1==1)
            {
            sensorstatus=0x01;
            }
            else
            {
            sensorstatus=0x00;
            }
}
void obstacledetection_chkleft()
{
            if(RA2==1)
            {
            sensorstatus=0x01;
            }
            else
            {
            sensorstatus=0x00;
            }
}
void obstacledetection_chkfront()
{
            if(RA3==1)
            {
            sensorstatus=0x01;
            }
            else
            {
            sensorstatus=0x00;
            }
}
void obstacledetection_chkback()
{
```

```
        if(RA4==1)
        {
        sensorstatus=0x01;
        }
        else
        {
        sensorstatus=0x00;
        }
}
void rftransmitter()
{
                te = 0;
                delay(20600);
                delay(20600);
                te = 1;
                delay(10000);
}

void smclockwisecalc()
{
                if((smcount==8)||(smcount==0))
                {
                        smcount=1;
                }
                else
                {
                smcount=(smcount*2);
                }
}
void smanticlockwisecalc()
{
                if((smcount==1)||(smcount==0))
                {
                        smcount=8;
                }
                else
                {
```

```
                smcount=(smcount/2);
                }
}
void smoff()
{
                if((smcount==1)||(smcount==0))
                {
                        RD4=0;
                        RD5=0;
                        RD6=0;
                        RD7=0;
                }
}
void smrun()
{
if(smcount==1)
                {
                        RD4=1;
                        RD5=0;
                        RD6=0;
                        RD7=0;
                        delay(60000);
                        delay(60000);
                        smoff();
                }
                else if(smcount==2)
                {
                        RD4=0;
                        RD5=1;
                        RD6=0;
                        RD7=0;
                        delay(60000);
                        delay(60000);
                        smoff();
                }
                else if(smcount==4)
                {
```

```
                        RD4=0;
                        RD5=0;
                        RD6=1;
                        RD7=0;
                        delay(60000);
                        delay(60000);
                        smoff();
                }
                else if(smcount==8)
                {
                        RD4=0;
                        RD5=0;
                        RD6=0;
                        RD7=1;
                        delay(60000);
                        delay(60000);
                        smoff();
                }
}
//**************************************************
//          Check Error
//**************************************************

void interrupt rcx(void)
{
if(INTF==1)
{
INTE=0;
GIE=0;
INTF=0;
rf=0x01;
INTE=1;
GIE=1;
}
}
//**************************************************
//          Main function
```

```
//***************************************************

void main()
{
ADCON1=0x8e;
TRISE=0x84;
PORTE=0x84;
TRISA=0xff;
PORTA=0xff;
TRISB=0xff;
PORTB=0xff;
TRISC=0x00;
TRISD=0x00;
PORTC=0x00;
PORTD=0x00;
RBPU=0;
//lcd_init();
delay(100);
lcd_condis("Aircraft  Project",16);
i=0x00;
rf=0x00;
te = 1;
        GIE=1;
        PEIE=1;
        INTE=1;
        INTEDG=1;
        lcd_command(0xc0);
        delay(20000);
while(1)
{
if(rf!=0x00)
{
                count=0;
                lcd_command(0x87);
                rf=0x00;
                portdata =PORTB;
                portdata=(portdata>>4);
```

```
portdata=(portdata&0x0f);
portdata=(portdata+0x30);
lcd_data(portdata);
TXREG=portdata;
if(portdata==0x31)
{
standby();
}
else if(portdata==0x32)
{
upward();
delay(60000);
delay(60000);
delay(60000);
delay(60000);
standby();
}
else if(portdata==0x33)
{
downward();
delay(60000);
delay(60000);
delay(60000);
delay(60000);
standby();
}
else if(portdata==0x34)
{
        for(count=0;count<45;count++)
        {
        delay(4000);
        obstacledetection_chkfront();
        if(sensorstatus==0x00)
        {
                if(count>5)
                {
                reverse();
```

```
                    delay(60000);
                    delay(60000);
                    delay(60000);
                    }
          count=50;
          }
          else
          {
          forward();
          }
          }
standby();
}
else  if(portdata==0x35)
{
          for(count=0;count<45;count++)
          {
          delay(4000);
          obstacledetection_chkback();
          if(sensorstatus==0x00)
          {
                    if(count>5)
                    {
                    forward();
                    delay(60000);
                    delay(60000);
                    delay(60000);
                    }
          count=50;
          }
          else
          {
          reverse();
          }
          }
standby();
}
```

```
else if(portdata==0x36)
{
        for(count=0;count<45;count++)
        {
        delay(4000);
        obstacledetection_chkright();
        if(sensorstatus==0x00)
        {
                if(count>5)
                {
                left();
                delay(60000);
                delay(60000);
                delay(60000);
                }
        count=50;
        }
        else
        {
        right();
        }
        }
standby();
}
else if(portdata==0x37)
{
        for(count=0;count<45;count++)
        {
        delay(4000);
        obstacledetection_chkleft();
        if(sensorstatus==0x00)
        {
                if(count>5)
                {
                right();
                delay(60000);
                delay(60000);
```

```
                    delay(60000);
                    }
          count=50;
          }
          else
          {
          left();
          }
          }
standby();
}
else if(portdata==0x38)
{
fastforward();
delay(60000);
delay(60000);
delay(60000);
delay(60000);
delay(60000);
delay(60000);
delay(60000);
delay(60000);
standby();
}
else if(portdata==0x39)
{
fastreverse();
delay(60000);
delay(60000);
delay(60000);
delay(60000);
delay(60000);
delay(60000);
delay(60000);
delay(60000);
standby();
}
```

```
else if(portdata==0x3a)
{
adcconvert1();
adcconvert1();
PORTD=0x3c;
rftransmitter();
PORTD=sensor13;
rftransmitter();
PORTD=sensor12;
rftransmitter();
PORTD=sensor11;
rftransmitter();
obsdet=PORTA;
obsdet=(obsdet>>1);
obsdet=(obsdet&0x0f);
PORTD=obsdet;
rftransmitter();
}
else if(portdata==0x3b)
{
smclockwisecalc();
smrun();
}
else if(portdata==0x3c)
{
smanticlockwisecalc();
smrun();
}
else if(portdata==0x3d)
{
        do
        {
        smclockwisecalc();
        smrun();
        }while(RE2==0);

}
```

```
else if(portdata==0x3e)
{
        do
        {
        downward();
        delay(60000);
        delay(60000);
        standby();
        delay(60000);
        delay(60000);
        }while(RA5==1);
PORTC=0x00;
motor34_7v=0;
}

delay(600);
}
}
}
```

6.3 Limitations

The limitations of our project goes here. The flight of the helicopter can be improved by using more powerful brushless motors. Now it cannot lift much weight.

CHAPTER 7

Testing & Maintenance

7.1 Tests

7.1: Unit test chart

No	Unit Name	Test Status
1	Base Station	Complete
2	Flying Part	Partial
3	Code	Complete
4	User Interface	Complete

The hardware components were tested for connectivity using the multimeter and the voltage and current ratings were found out. Each unit circuits were tested separately and changes were made. Then the units were integrated one by one into the main module and the working was checked with the corresponding code. Necessary changes were made in the circuits and the code. At last the entire system was tested.

7.2 Maintenance

The helicopter and its circuit has to be maintained properly. Necessary changes in the circuit and code should be made to adapt to the changes. Improved or effective components may be used to enhance the performance according to the various needs.

CHAPTER 8
Conclusion

8.1 Introduction

As explained earlier our project is controlling a helicopter which we constructed using a PC or laptop. The direction of the helicopter can be controlled. There is an RF camera mounted on the helicopter which can be rotated 360 degrees using the same user interface. Video captured by the camera is viewed on the screen of controlling PC. The temperature and direction are sensed by different sensors and the details are passed to the base station.

8.2 Future work

In the future the helicopter can be modified. Brushless motors can be used to improve the flight performance and weight balancing. More weight can be made to uplifted. Other features can be added in the system. More sensors can be added.

REFERENCES

[1] Melonee Wise,*Wireless Computer Controlled Robotics Using PIC16F77 Microcontroller*

[2] Gooligum Electronics,*Introduction to PIC Programming by David Meiklejohn*

[3] Yesu Thommandru,*Programming a PIC Microcontroller-A Short Tutorial*

[4] Ainsmar X. Brown and Richard D. Garcia,*Control of an Autonomous Radio-Controlled Helicopter in a Modified Simulation Environment Using Proportional Integral Derivative Algorithms*

[5] Xiaoyun Zhu,Michiel van Nieuwstadt Control,*"The Caltech Helicopter Control Experiment"*

[6] Sanghoon Cha Brown University, scha@cs.brown.edu,*Sanghoon Cha Brown University, scha@cs.brown.edu*

[7] *Computer based control system for a model helicopter*

[8] Steven Northrup and Christopher Paros, Department of Electrical Engineering, Western New England College, Springfield, MA 01119, snorthru@ wnec.edu,*Work in Progress - Capstone Experience - Visual Navigation of an RC Vehicle using Wireless Video Feedback to a PC*

[9] Yuta Yoshihata, Kei Watanabe, Yasushi Iwatani and Koichi Hashimoto, *Multi-Camera Visual Servoing of a Micro Helicopter Under Occlusions*

[10] CAI Jun,YU Shun-Zheng and LIU Jing-li,*The Design of a Wireless Data Acquisition and Transmission System*

[11] Carlo Ghezzi, *Fundamentals of software engineering*,2nd Edition